

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2003-6996

(P2003-6996A)

(43)公開日 平成15年 1月10日 (2003. 1. 10)

(51)Int.Cl. ⁷	識別記号	F I	テーマコード [*] (参考)
G 1 1 B 20/12		G 1 1 B 20/12	5 D 0 4 4
20/18	5 5 0	20/18	5 5 0 E
	5 6 0		5 6 0 G
	5 7 2		5 7 2 B
			5 7 2 C

審査請求 未請求 請求項の数 5 O L (全 12 頁) 最終頁に続く

(21)出願番号 特願2001-185334(P2001-185334)

(22)出願日 平成13年 6月19日 (2001. 6. 19)

(71)出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中 4丁目 1番
1号

(72)発明者 内田 好昭

神奈川県川崎市中原区上小田中 4丁目 1番
1号 富士通株式会社内

(74)代理人 100072590

弁理士 井桁 貞一

Fターム(参考) 5D044 AB01 BC01 BC06 CC04 DE02

DE03 DE68 DE83 GK12 GK19

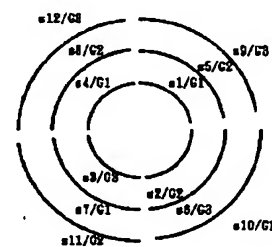
(54)【発明の名称】 ディスク読み書き制御プログラム

(57)【要約】

【課題】 ディスク装置のデータの読み出し時の訂正不能なエラー発生確率を低くし、これによって書き込み時の検証処理をも省略可能とすることを目的とする。

【解決手段】 所定数Lバイトのデータを単位(セクタ)として、所定数n個のデータセクタを一つのグループとして管理して記憶媒体への書き込みと読み出しとをコンピュータに実行させるディスク読み書き制御プログラムであって、コンピュータを同一グループのデータセクタから各1バイトずつ抽出したnバイトのデータに対してmバイトの誤り検出訂正符号を生成する処理をLバイトの全データ分繰り返し、Lバイトからなるm個のパリティセクタを形成するパリティ生成手段と、同一グループのn個のデータセクタと対応するm個のパリティセクタとを一つのパリティグループとして、同一のパリティグループに属するセクタが隣接しないように分散配置して記憶媒体に書き込む読み書き制御手段として機能させる。

セクタ配置例



例 トラック当たりのセクタ数: $N=4$
一つのパリティグループを構成するセクタ配置間隔: $k=3$
セクタ番号: s_{ij}
パリティグループ番号: C_j
パリティグループ1のセクタ: —

【特許請求の範囲】

【請求項1】 所定数 L バイトのデータを単位（セクタ）として、所定数 n 個のデータセクタを一つのグループとして管理して記憶媒体への書き込みと読み出しとをコンピュータに実行させるディスク読み書き制御プログラムであって、コンピュータを同一グループのデータセクタから各1バイトずつ抽出した n バイトのデータに対して m バイトの誤り検出訂正符号を生成する処理を L バイトの全データ分繰り返し、 L バイトからなる m 個のパリティセクタを形成するパリティ生成手段と、同一グループの n 個のデータセクタと対応する m 個のパリティセクタとを一つのパリティグループとして、同一のパリティグループに属するセクタが互いに物理的に隣接しないように分散配置して記憶媒体に書き込む読み書き制御手段として機能させるためのディスク読み書き制御プログラム。

【請求項2】 所定数 L バイトのデータを単位（セクタ）として、所定数 n 個のデータセクタを一つのグループとして管理して記憶媒体への書き込みと読み出しとを行うディスク読み書き制御装置であって、同一グループのデータセクタから各1バイトずつ抽出した n バイトのデータに対して m バイトの誤り検出訂正符号を生成する処理を L バイトの全データ分繰り返し、 L バイトからなる m 個のパリティセクタを形成するパリティ生成手段と、同一グループの n 個のデータセクタと対応する m 個のパリティセクタとを一つのパリティグループとして、同一のパリティグループに属するセクタが互いに物理的に隣接しないように分散配置して記憶媒体に書き込む読み書き制御手段とを有することを特徴とするディスク読み書き制御装置。

【請求項3】 請求項1に記載のディスク読み書き制御プログラムを用いて書き込まれたデータを読み出すディスク読み書き制御プログラムであって、読み書き制御手段は、データセクタ読み出し時にセクタ内で訂正できないエラーを検出した場合は、パリティセクタを読み出してエラー訂正を行うことを特徴とするディスク読み書き制御プログラム。

【請求項4】 請求項1に記載のディスク読み書き制御プログラムであって、記憶媒体上を所定の領域に分割し、個々の領域ごとにデータの読み出しエラー率、およびシークエラー率を計測するエラー率計測手段を有し、読み書き制御手段は、該エラー率が所定の値よりも大きい場合は、書き込み後の検証処理を行なうことを特徴とするディスク読み書き制御プログラム。

【請求項5】 請求項1に記載のディスク読み書き制御プログラムを用いてデータが書き込まれたコンピュータ読み取り可能な記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、ディスク装置のデ

ータ書き込み／読み出し制御に関し、特に、データの読み出し時の誤り訂正能力を高めるデータ書き込み／読み出し制御プログラムに関する。

【0002】

【従来の技術】 フロッピー（登録商標）ディスク、光磁気ディスク等の記憶装置では、記憶媒体上に、図15に示すごとく同心円状にデータを記録するトラックが設けられ、各トラックは複数のデータの単位（セクタ）に分割されてデータが記録される。例えば、直径90mmの光磁気ディスクで記憶容量540MBの場合、トラック間隔：1.1 μ m、セクタ長：4mm程度である。通常の事務所環境では、直径数 μ m～10 μ m程度迄のゴミがあるが、このようなゴミが、図15に示すごとく記憶媒体表面に付着した場合、読み出しエラーが発生する。このエラー対策としては、以下の方法がある。

【0003】 1. データ書き込み直後にデータを読み出し、書き込んだデータと読み出したデータを照合するバリファイ（検証）を行い、正しいデータが読み出せない場合は、再度、書き込み／読み出し／検証を行い、データが正しく書き込まれたことを確認する。

2. 書き込み後に発生するエラー対策として、誤り検出／訂正手段（ECC手段）を設け、セクタごとに誤り検出訂正符号を付加して記録する。この方法では、データ読み出し時に発生する（ECC手段の誤り訂正能力範囲内の）多少のエラーについては、ECC手段により訂正して出力する。

【0004】 しかし、これらの方法には、各々、以下に示す問題がある。

1. データ書き込み直後に検証を行う方法は、データ書き込み時に同一箇所の、書き込みと検証のための読み出しを行うため、少量のデータ書き込みであっても書き込み後の検証処理のための記憶媒体回転待ちを伴い、正味の書き込み時間と比較して、実際の書き込み時間が長くなる。

【0005】 2. 一つのセクタ内で、ECC手段の誤り訂正能力を超えるエラーが発生した場合は、やはり読み出しエラーとなる。この問題の対策例として、例えば

「特開平5-89609」が公知である。掲記公知例では、図16の表に示すごとく、各データセクタのバイト位置を揃えて配列した所定のグループのデータから、同一位置のバイトを抽出したバイト列を生成し、データセクタと生成したバイト列に対して生成したパリティ符号または誤り検出訂正符号で構成されたセクタ、すなわちパリティセクタとをひとつのパリティグループとして記憶媒体に書き込む。

【0006】 このようにすることで、あるセクタでECC手段の誤り訂正能力を超えるエラーが発生した場合でも、エラーが発生したデータセクタの各バイトの値を、同一のパティグループの他のデータセクタとパリティセクタの当該バイト位置の値とから逆算して訂正可能とな

る。しかし、掲記公知例の方法では、一つのパリティグループを構成するセクタの、記憶媒体上での物理的な配置を考慮していないため、一つのパリティグループを構成するセクタが物理的に隣接して配置される場合がある。

【0007】しかし、直径数 μm ~10 μm 程度迄のゴミが記憶媒体表面に付着した場合、このゴミは、隣接するセクタ、あるいは1 μm 程度の間隔で隣接するトラックにまたがり、この範囲でデータセクタに付加した誤り検出訂正符号だけでは訂正できないエラーを発生させる。図17は、一つのパリティグループを構成するセクタを連続してセクタに配置した従来例にゴミが付着し、セクタに付加した誤り検出訂正符号だけでは訂正できないエラーを発生させた例を示す。この場合、一つのパリティグループの第一セクタと第二セクタとにまたがってゴミが付着し、第一セクタと第二セクタとで訂正不能なエラーが同時に発生し、パリティセクタを用いても誤り訂正不能となる。

【0008】図18は、図17と同様のセクタ配置の記憶媒体の、複数のトラックにまたがってゴミが付着した例を示す。この場合も、図17の例と同様に誤り訂正不能となる。

【0009】

【発明が解決しようとする課題】本発明は、データの読み出し時の訂正不能なエラー発生確率を低くし、これによって書き込み時の検証処理をも省略可能とすることを目的とする。

【0010】

【課題を解決するための手段】請求項1に記載の発明は、所定数 L バイトのデータを単位（セクタ）として、所定数 n 個のデータセクタを一つのグループとして管理して記憶媒体への書き込みと読み出しとをコンピュータに実行させるディスク読み書き制御プログラムであって、コンピュータを同一グループのデータセクタから各1バイトずつ抽出した n バイトのデータに対して m バイトの誤り検出訂正符号を生成する処理を L バイトの全データ分繰返し、 L バイトからなる m 個のパリティセクタを形成するパリティ生成手段と、同一グループの n 個のデータセクタと対応する m 個のパリティセクタとを一つのパリティグループとして、同一のパリティグループに属するセクタが互いに物理的に隣接しないように分散配置して記憶媒体に書き込む読み書き制御手段として機能させる。

【0011】請求項2に記載の発明は、所定数 L バイトのデータを単位（セクタ）として、所定数 n 個のデータセクタを一つのグループとして管理して記憶媒体への書き込みと読み出しとを行うディスク読み書き制御装置であって、同一グループのデータセクタから各1バイトずつ抽出した n バイトのデータに対して m バイトの誤り検出訂正符号を生成する処理を L バイトの全データ分繰

返し、 L バイトからなる m 個のパリティセクタを形成するパリティ生成手段と、同一グループの n 個のデータセクタと対応する m 個のパリティセクタとを一つのパリティグループとして、同一のパリティグループに属するセクタが互いに物理的に隣接しないように分散配置して記憶媒体に書き込む読み書き制御手段とを有することを特徴とする。

【0012】これにより、ゴミの付着等で物理的な局所にまとまったデータエラーが発生しても、同一パリティグループのセクタに訂正不能なエラーが集中して発生する確率が低くなる。したがって、書き込み時の検証処理が省略可能となる。セクタが同心円状に配置され、トラックあたりのセクタ数が N の記憶媒体に対し、同一パリティグループのセクタを物理的に分散配置する方法として、 $(n+m)$ を N に等しいか小さくする方法がある。これにより、同一パリティグループのセクタが隣接するトラックの同一位置に配置されることを防止できる。

【0013】同一パリティグループのセクタを物理的に分散配置する他の方法として、以下に示す方法を用いても良い。 N と互いに素な数 k によって、 k セクタごとに一つのパリティグループのセクタを配置する。すなわち、セクタ番号 $s = i, i+k, i+2 \times k, i+3 \times k, \dots, i+(n+m-1) \times k$ を一つのパリティグループとする。

【0014】図1は、 $N=4$ 、 $k=3$ とした場合のセクタ配置例を示す。ここでは、内周から順に $s_1 \sim s_{12}$ までセクタ番号が付与されており、 s_1 、 s_4 、 s_7 、 s_{10} が、パリティグループ1を構成する。請求項3に記載の発明は、請求項1に記載のディスク読み書き制御プログラムを用いて書き込まれたデータを読み出すディスク読み書き制御プログラムであって、読み書き制御手段は、データセクタ読み出し時にセクタ内で訂正できないエラーを検出した場合は、パリティセクタを読み出してエラー訂正を行うことを特徴とする。

【0015】すなわち、読み出したデータセクタにエラーがない場合、あるいは、読み出したデータのセクタに誤り検出訂正符号が付加されていて、その内容だけでエラー訂正が可能な場合は、パリティセクタの読み出しを省略する。したがって、訂正不能なエラーが生じない限り、データ読み出しに要する時間はパリティセクタを設けなかった場合と同じままである。訂正不能なエラー発生時は、より強力なエラー訂正対応が可能となる。

【0016】請求項4に記載の発明は、請求項1に記載のディスク読み書き制御プログラムであって、記憶媒体上を所定の領域に分割し、個々の領域ごとにデータの読み出しエラー率、およびシークエラー率を計測するエラー率計測手段を有し、読み書き制御手段は、該エラー率が所定の値よりも大きい場合は、書き込み後の検証処理を行なうことを特徴とする。

【0017】これにより、記憶媒体表面への異物の付着

による一時的なエラーだけでなく、異物の固着等による表面状態の変化に起因する読み出しエラーに対して動的にエラー対策を更新して訂正不能なエラーの発生確率を低くすることが可能となる。

【0018】

【発明の実施の形態】図2は、ハードディスク（HDD）1のデータを光磁気ディスク（MO）7に高速に複写する実施例1の構成例を示す。読み書き制御手段5は、上位装置等からの指示により、バッファ2に対しHDD1のデータ読み出しを指示し、バッファ2は、HDD1から読み出したデータを一時蓄える。

【0019】パリティ生成手段4は、読み書き制御手段5からの指示にしたがって、パリティバッファ3を初期化し、バッファ2がHDD1から1セクタ分のデータを読み込む毎に、読み込んだデータを参照して新たなパリティを生成し、パリティバッファ3のデータを更新する。パリティバッファ3は、その時点で生成しているパリティデータを蓄えるもので、一つの処理で扱うパリティグループの数を g とすると、 g セクタ分のバッファを有する。なお、バッファ2はパリティデータを作成するための作業域としても用いるため、パリティバッファ3と同程度の容量を持つことが望ましい。

【0020】読み書き制御手段5は、HDD1からバッファ2に読み込んだデータとパリティバッファ3のデータとをMO7に書き出し、パリティバッファを初期化（ゼロクリア）する。実施例1によって図1に示すパリティグループデータの記憶媒体上でのセクタ配置例を実現し、 $n=3$ 、 $m=1$ として、ここに図6に示すごみが付着した例の処理を説明する。この例では、パリティグループG1を構成する3個のデータセクタを、各々、 s_1 、 s_4 、 s_7 に、各データセクタのデータから生成した1個のパリティセクタを、 s_{10} に書き込む。

【0021】概略処理手順は、1. HDD1から読み込んだ所定の3セクタ分のデータをパリティグループとしてバッファ2に保持する。2. パリティ生成手段4は、所定の3セクタの対応する n バイトの排他論理和（xor）をとりパリティセクタを生成し、パリティバッファ3に送る。3. 読み書き制御手段5は、バッファ2の3セクタ分のデータと、生成したパリティセクタとを、各々、 s_1 、 s_4 、 s_7 、 s_{10} に書き込む。

【0022】これにより、図6に示すような、複数のトラックにまたがるゴミが付着し、セクタ単位では訂正不能のデータエラーを複数のセクタで生じる場合でも、誤り訂正不能なセクタは、データを書き込んだパリティグループG1でみると、 s_1 セクタのみである。したがって、 s_4 、 s_7 、 s_{10} セクタの各バイトを用いて s_1 セクタのエラー訂正が可能となる。

【0023】エラー率計測手段6は、読み書き制御手段5から読み出し時の検証情報と読み出したデータの位置情報とを受け取り、当該位置におけるエラー発生状況か

らMO7に対する読み書きに伴うエラー率を計算し、パリティ生成手段4は、該エラー率にしたがって n の値（一つのパリティグループの大きさ）を更新する。例えば、エラー率が高い場合は n を小さく、エラー率が低い場合は、 n を大きく設定する。

【0024】計算したエラー率をMO7に書き出して記憶媒体に保存しておき、記憶媒体挿入時にこの情報を読み出し、処理するようにしてもよい。パリティグループの数 g はMOディスク毎に固定したものでもよいし、MOディスクのゾーン毎などによって変化してもよい。この例では、ある時点でのパリティグループの数を g とし、 n セクタのデータに対して、 m セクタのパリティデータを生成・付与するものとする。

【0025】データを書き込む際の処理の概略フローを図3に示す。読み書き制御手段5は、上位装置からの書き込み要求にしたがってシーク（位置づけ）動作を行ない、シークエラーが発生した場合は再実行する。さらに、エラー率計測手段6は、シーク動作のエラー回数をカウントする。データを読み出す際の処理の概略フローを図4に示す。エラー率計測手段6は、データ書き込みの際と同様にエラー回数をカウントする。

【0026】エラー率計測手段6は、上記のシーク動作や読み出し動作におけるシークエラー、および読み出しエラーの発生頻度をそのセクタ番号とともに記憶し、読み出しエラー率が所定の値よりも大きくなった場合には、次のいずれかのエラー処理を行なう。

(1)読み書き制御手段5は、書き込み後の検証を強制的に行なう。

【0027】(2)読み書き制御手段5は、書き込み後の検証を所定回数実施し、読み出しエラーがなくならない場合はセクタの交代処理を行う。

(3)パリティ生成手段4は、一つのパリティグループを構成するセクタ数 n を小さくする、あるいは、一つのパリティグループに対して生成する誤り検出訂正符号を多くすることで、誤り訂正能力を高める。

【0028】上記の(1)は比較的簡便な処理であり、従来の制御との差異が小さいので実装が容易である。(2)は、「当該位置ではエラーが頻発するので再実行してもエラーになる可能性が大きい」と判断し、セクタの交代を行う。(3)はエラー率が大きくなる程度が数倍程度の範囲と予想し、それに見合う程度に誤り訂正能力を高めるものである。

【0029】なお、誤り検出訂正符号を、たとえば「情報のデジタル伝送」森北出版 ISBN4-627-82350-9、「コンピュータ基礎講座 18 符号理論」宮川洋ほか、昭晃堂等、に記述されているリード・ソロモン符号とすることで、より強力な訂正が可能となる。また、パリティ生成手段4は、「セクタ単位のエラー検出・訂正により、どのセクタでエラーが生じているかをほぼ特定できる」という特性を利用して、独立な m 個の計算式によって m 個

のパリティセクタを計算することができる。すなわち、何れのm項を取り除いても互いに一次独立となるようなm個の式によってパリティセクタの値を求めておく。これによって、いずれかのセクタでエラーが生じた場合、そのセクタの値を未知とする関係式をm個構築でき、これを解くことでエラーが発生したセクタの値が求められる。

【0030】以上述べた方法でパリティセクタを生成し、読み出しエラー発生時にはパリティセクタを読み出してエラー訂正が可能となる。エラー率にしたがってエラー処理を行う疑似コード例を図5に示す。図7は、デジタルビデオ(DV)機器等からの、リアルタイム性が要求されるデータを読み込み、MOに書き出す実施例2の構成例を示す。リアルタイム性が要求される場合、書き込み後の検証／再書き込みが発生すると、最悪値として保証すべき書き込み時間を超える可能性があるが、本発明を適用し、書き込み後の検証を省略し、上記可能性を回避できる。

【0031】この例ではDV11からの入力データの取り込みをDMA処理とし、パリティグループの大きさnは固定している。いずれも、並列処理を容易にし、より高速処理が可能となる。また、パリティ生成論理は、xorなどの簡単な論理だけでなく $m > 1$ となるリードソロモン符号のごとく、エラー訂正能力の高い論理を用いてもよい。

【0032】図8は、実施例2のエラー率計測手段16の内部構成と動作概念図を示し、DV11からのデータの取り込み(MO17への書き込み)が終了したときのデータ検証動作について説明する。図8において、エラーマップ62は、訂正不能な読み出しエラーが発生するセクタの場所を示すテーブル、もしくはビットマップである。パリティセクタアドレス61はデータを保存したときのパリティグループの数gとパリティグループを構成するセクタ数nとを保持し、この値をMO17に書き込む。

【0033】読み書き制御手段15は、直前に書き込んだデータをMO17から読み出し、バッファ12に蓄える。このとき読み出し時に訂正できないエラーが発生すると、エラーマップ62にその旨記録する。 $g \times (n + m)$ セクタの読み出しが終了したとき、エラー訂正手段63が起動する。エラー訂正手段63は、エラーマップ62を参照し、訂正不能な読み出しエラーが記録されているとき、図9に示す、エラー訂正処理を試みる。

【0034】このように処理することにより、読み出し不可能なセクタが残る確率を極めて小さなものにすることができる。なお、図8の処理については、実行の開始指示手段を設けるとともに、実行中であることを表示し、オペレータの指示によって、処理を開始するようにしてもよい。

【0035】また、図8の処理により、正しく読み出せ

ないセクタが見つかり、エラー訂正処理でそのセクタを書き戻しても読み出しエラーとなる場合がある。この場合、通常は二次欠陥リスト(SDL)から交代セクタが割当てられて、交代セクタに書き込まれる。交代セクタ領域を使い切っているために交代セクタを割当てることのできない場合はエラーとしてオペレータに通知するようにしてもよい。

【0036】図10は、本発明を適用し、エラー率が所定の値よりも大きい場合に所定数nを減少させる手段として、パリティセクタ領域①と、二次パリティ領域および二次パリティ管理②と、エラー率管理領域③とを備えた実施例3のファイルシステムの、ディスク内の論理データのレイアウト例を示す。実施例3においては、パリティセクタはデータ領域のnセクタからなるクラスタ単位、すなわち、ひとつのパリティグループ毎に付与される。パリティセクタ領域①のパリティセクタ数は、概ね32000~64000セクタ程度となる。二次パリティセクタ領域は、より小さな単位に対して付与したパリティセクタの領域であり、エラーの発生状況に応じて使われるので、どの部分に対して二次パリティセクタを付与したかを管理する二次パリティ管理②を設けている。エラー率管理領域③は、記憶媒体上のエラー多発領域の位置を保持する。

【0037】このファイルシステムでは、ディスクへのデータ書き込みの際、クラスタ単位でパリティデータを作成し、これをパリティセクタ領域①に書き込む。さらに、二次パリティ管理②を参照して、エラーが多発する領域であれば、例えばクラスタの前半部分についてのパリティを求めて、これを二次パリティセクタ領域に書き込む。これは、実質的にnを1/2にしたことに相当する。

【0038】二次パリティセクタ処理の疑似コード例を図11に示す。図中、①の処理は、8KByte(=64Kbit)のbitmap、すなわち2次パリティセクタを記録すべきクラスタならば'1'、そうでなければ'0'のビットマップを行う。②の処理は、二次パリティセクタの対象位置、すなわち「クラスタ番号」(2Byte)と「クラスタ内位置」「セクタ数」(各4ビット)の構造体の配列を行う。{1203, 4, 4}であればクラスタ1203先頭から4/16の部分から4セクタ分のパリティを作っていることを示す。

【0039】③の処理は、二次パリティセクタ、すなわち②で指される部分のパリティを求める。データ読み出しは、次の点を除いて、概ね従来のファイルシステムとして動作する。すなわち、

(1) 読み出しエラーが発生した時、その場所と頻度を計数して、エラー率管理部に登録する。

(2) 訂正不能な読み出しエラーが発生したとき(通常のファイルシステムはそのままエラーを通知するが)エラー発生場所をシステム情報として記録しておく。これ

により、その直後に起動したソフトウェアがエラーの発生場所を知って、対応するパリティ情報などからエラーを修復することができる。

【0040】なお、ここではFATファイルシステムを想定したが、クラスタやアロケーション単位に相当する概念を持つものである限り、他のファイルシステムでも同様の構成ができる。図12に、実施例3のクラスタデータの書き込み処理フロー例を示す。実施例4として、光磁気ディスク装置に、装置自身の機能として本発明を適用する例を説明する。230MByte～640MByteのMOディスクはZCLVと呼ばれる方式を採用している。これはディスクの記録領域を10～17個のゾーンに分けて、ゾーン毎にトラック当りのセクタ数を変えるものである。内周ではトラック当りのセクタ数は少なく、外周にいくほど多くなる。例えば540MByteのMOディスクでは18個のゾーンからなる。最内部は875トラックでトラック当り50セクタであり、最外部は875トラック、トラック当り84セクタである。

【0041】各ゾーン毎にトラック当りのセクタ数Nに基づいて、 $k = N - 3$ セクタおきにひとつのパリティグループとする。すなわち、最内部についてはセクタ番号sのセクタは $(s \bmod 47)$ 番のパリティグループとする。各パリティグループ毎にひとつのパリティセクタを用意して、データ書き込み時にはこの各パリティグループ毎にパリティ値を求めて、パリティセクタを書き込む。

【0042】実施例4のセクタ配置の概念図を図13に示す。図においては、記憶媒体上でひとつのパリティグループとして配置したセクタのみを線で示している。例えば、ハッチした部分の幅はおよそ20 μm 、すなわち20トラック分であるが、ここに10 μm 程度のゴミが付着しても、この10トラック分の円周方向10 μm の領域に配置される、同一パリティグループに属するセクタは1セクタ以内にとどまり、エラー訂正が可能である。

【0043】また、実施例4においては、ゾーン毎に同じゾーンのセクタについてのパリティ値を保存するパリティセクタの領域を設ける。最内部の例ではkは $N - 3 = 47$ セクタである。当該ゾーン内のパリティグループの大きさは、総セクタ数/ $k = 875 \times 50 / 47 = 981$ であり、パリティセクタ数は47となる。更に、個々のパリティグループについて各セクタの状態を管理する領域を設ける。各セクタについて{解放状態/書き込み済み/上書きされた}の3状態を管理する。状態管理のためには2ビットの情報があれば十分であるため、パリティグループ毎に1セクタ、すなわちk個の状態管理セクタがあればよい。

【0044】従来のディスクと互換を持たせる場合は、低レベルフォーマット時に2k個の欠陥セクタがあるかのように初期欠陥リスト(PDL)を登録する。この欠

陥セクタとして登録したセクタをパリティセクタとして利用する。以下、説明のために、リニアナンバリングされた「セクタ番号s」が $k \times j + (i - 1)$ のとき、このセクタは「パリティグループi」に属しており、「パリティセクタi」と「状態管理セクタi」で管理されているものとする。

【0045】実施例4のデータ書き込み手順を図14に示す。例えば画像データなどの大容量データを書き込むとき、ユーザアプリケーションプログラムの動作例を以下に示す。

1. 予めディスクを初期化しておく。
2. 書き込むデータの大きさを(余裕をみて)見積もり、その分の領域をディスク上に確保し、必要に応じて領域を解放する。見積もり方法については、たとえば動画データのおよその時間とビットレートから推定する。
3. データを逐次書き込む。
4. ファイルをクローズする。
5. ディスクを排出する。

【0046】これに伴い、内部では次のように処理する。

1. 状態管理セクタとパリティセクタを初期化する。
2. 書き込むゾーンの状態管理セクタとパリティセクタを初期化する。
3. 書き込みに従って状態管理セクタを更新し、パリティセクタを更新する。ゾーンの切り替えが発生した場合はこれらをディスクに書き出す。
4. バッファに保持しているパリティセクタと状態管理セクタをディスクに書き出す。

【0047】上の処理では、状態管理セクタの内容が{上書き済み}を示すことはない。このように書き込んだデータについて、ゴミの付着等により書き込み後にデータが読めなくなっても、それがパリティグループ内に1セクタに留まるかぎり、他のセクタの値から復元することが可能である。なお、ここでは説明の都合で一つのゾーンを一括管理するように説明したが、kをより大きくしたり、ゾーンを前半部・後半部に細分することで一つのパリティグループを小さくすることができる。

【0048】実施例5として、記録媒体上のセクタ長のn倍、すなわち、ひとつのパリティグループを論理的なセクタ長としてみせる記憶装置としても良い。例えば、セクタ長が512Byteや2KByteの記憶媒体を用いる記憶装置において、複数のセクタを一パリティグループとして管理し、ホスト側にはセクタ長を例えば16KByteや32KByteとして通知するものを作ることができる。この一パリティグループ毎に1つ以上のパリティセクタを用意し、本発明を適用することができる。一般に、パリティグループとしてまとめたデータセクタとパリティセクタとの位置関係は任意である。例えば、パリティグループとしてまとめたセクタと連続した場所においてもよいし、管理用のブロックなどにパリティセクタだけをま

とめてもよい。

【0049】実施例6として、ハードディスクのデータを一括してバックアップするアプリケーションソフトとしても良い。例えば、バックアップソフトでは、まとまった量のデータを一括して書き込む。また、このソフトではバックアップ、すなわち書き込みの時間が重要な要素である。このようなソフトウェアでは、独自のファイル管理を行なうことができる。また、バックアップファイル毎に独自のパリティデータを生成して、これを書き込むこともできる。例えば、バックアップの書き込みが n セクタに達する毎に m セクタ分のパリティデータを生成して、これを書き込むとしても良い。

【0050】実施例7として、ファイルシステム、とくにデータ圧縮を伴うファイルシステムとしても良い。例えば、FAT16形式のファイルシステムにおいては、ディスクの領域をクラスタと呼ぶ単位で管理している。ディスク容量が508MByte程度るとき、クラスタの大きさは8KByteであり、セクタ長が512Byteであれば16セクタとなる。したがって、ディスクをフォーマットしてファイルシステムを構築する際に、全体の1/16程度の領域をパリティセクタとして割当てる。以後、ファイルを書き込むたびに（この書き込み処理はクラスタ単位で扱えるので）クラスタ毎にパリティセクタのデータを生成して、これを別途割当てた領域に書き込む。

【0051】データ圧縮を行なうことによってみかけのディスク容量を物理的なディスク容量よりも大きく見せるファイルシステムでは、8KByte～32KByte程度の大きさのデータを一単位として圧縮し、これを物理的にはより小さな領域に書き込んで管理することが可能となる。

（付記1）所定数 L バイトのデータを単位（セクタ）として、所定数 n 個のデータセクタを一つのグループとして管理して記憶媒体への書き込みと読み出しとをコンピュータに実行させるディスク読み書き制御プログラムであって、コンピュータを同一グループのデータセクタから各1バイトずつ抽出した n バイトのデータに対して m バイトの誤り検出訂正符号を生成する処理を L バイトの全データ分繰返し、 L バイトからなる m 個のパリティセクタを形成するパリティ生成手段と、同一グループの n 個のデータセクタと対応する m 個のパリティセクタとを一つのパリティグループとして、同一のパリティグループに属するセクタが互いに物理的に隣接しないように分散配置して記憶媒体に書き込む読み書き制御手段として機能させるためのディスク読み書き制御プログラム。

【0052】（付記2）所定数 L バイトのデータを単位（セクタ）として、所定数 n 個のデータセクタを一つのグループとして管理して記憶媒体への書き込みと読み出しとを行うディスク読み書き制御装置であって、同一グループのデータセクタから各1バイトずつ抽出した n バイトのデータに対して m バイトの誤り検出訂正符号を生

成する処理を L バイトの全データ分繰返し、 L バイトからなる m 個のパリティセクタを形成するパリティ生成手段と、同一グループの n 個のデータセクタと対応する m 個のパリティセクタとを一つのパリティグループとして、同一のパリティグループに属するセクタが互いに物理的に隣接しないように分散配置して記憶媒体に書き込む読み書き制御手段とを有することを特徴とするディスク読み書き制御装置。

【0053】（付記3）請求項1に記載のディスク読み書き制御プログラムを用いて書き込まれたデータを読み出すディスク読み書き制御プログラムであって、読み書き制御手段は、データセクタ読み出し時にセクタ内で訂正できないエラーを検出した場合は、パリティセクタを読み出してエラー訂正を行うことを特徴とするディスク読み書き制御プログラム。

【0054】（付記4）請求項1に記載のディスク読み書き制御プログラムであって、記憶媒体上を所定の領域に分割し、個々の領域ごとにデータの読み出しエラー率、およびシークエラー率を計測するエラー率計測手段を有し、読み書き制御手段は、該エラー率が所定の値よりも大きい場合は、書き込み後の検証処理を行なうことを特徴とするディスク読み書き制御プログラム。

【0055】（付記5）請求項1に記載のディスク読み書き制御プログラムを用いてデータが書き込まれたコンピュータ読み取り可能な記録媒体。

（付記6）請求項1に記載のディスク読み書き制御プログラムであって、記憶媒体上を所定の領域に分割し、個々の領域ごとにデータの読み出しエラー率、およびシークエラー率を計測するエラー率計測手段を有し、パリティ生成手段は、該エラー率が所定の値よりも大きい場合は、 m/n を増加させる、すなわち所定数 n を減少させる、あるいは所定数 m を増加させることを特徴とするディスク読み書き制御プログラム。

【0056】（付記7）請求項1に記載のディスク読み書き制御プログラムであって、セクタが同心円状に配置され、トラックあたりのセクタ数が N の記憶媒体に対し、読み書き制御手段は、 N と互いに素な数 k によって、 k セクタごとに同一パリティグループのセクタを分散配置して記憶媒体に書き込むことを特徴とするディスク読み書き制御プログラム。

【0057】

【発明の効果】以上説明したごとく、本発明によれば、1. 従来の記憶装置と同一、または同等のハードウェアを用いて、書き込み速度がより高速で、かつより安定している記憶装置とすることができる。なお、このためのオーバーヘッドは、パリティセクタとして利用されるディスクの記憶容量、すなわち全ディスクの記憶容量の1%-数%程度である。

2. 誤り訂正能力が高まり、書き込み後の検証が省略可能となり、データの書き込み処理の高速化が実現でき

る。また、書き込み後の検証を省略することで実質的な書き込み速度が安定し、ホストやアプリケーションに対して保証できる書き込み速度も高速となる。このことにより、記憶装置の用途を広げることが可能となる。

【図面の簡単な説明】

【図 1】セクタ配置例

【図 2】実施例 1 の構成例

【図 3】実施例 1 のデータ書き込み処理フロー例

【図 4】実施例 1 のデータ読み出し処理フロー例

【図 5】エラー率に従ってエラー処理を行う疑似コード例

【図 6】実施例 1 のセクタ配置とゴミの付着例

【図 7】実施例 2 の構成例

【図 8】実施例 2 のエラー率計測手段の内部構成と動作概念図

【図 9】エラー訂正処理例

【図 10】実施例 3 のディスク内の論理データのレイアウト例

【図 11】二次パリティセクタ処理の疑似コード例

【図 12】実施例 3 のクラスタデータの書き込み処理フロー例

【図 13】実施例 4 のセクタ配置の概念図

【図 14】実施例 4 のデータ書き込み手順

【図 15】同心円状にデータを記録するトラックとセクタ例

【図 16】パリティセクタ生成例

【図 17】連続してセクタを配置した従来例にゴミが付着した例 1

【図 18】連続してセクタを配置した従来例にゴミが付着した例 2

【符号の説明】

c、d：シークリトライ回数

g：一つの処理で扱うパリティグループ数

G：パリティグループ番号

k：一つのパリティグループを構成するセクタを配置する間隔

m：一つのパリティグループに付加するパリティセクタ数

n：一つのパリティグループを構成するデータセクタ数

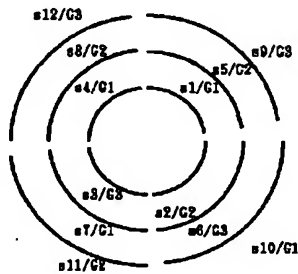
N：一トラック上のセクタ数

L：セクタ当たりのデータ長

s：セクタ番号

【図 1】

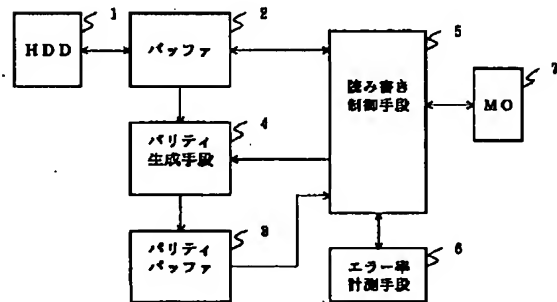
セクタ配置例



例 トラック当たりのセクタ数：N=4
一つのパリティグループを構成するセクタ配置間隔：k=8
セクタ番号：sxx
パリティグループ番号：Gy
パリティグループ 1 のセクタ：—

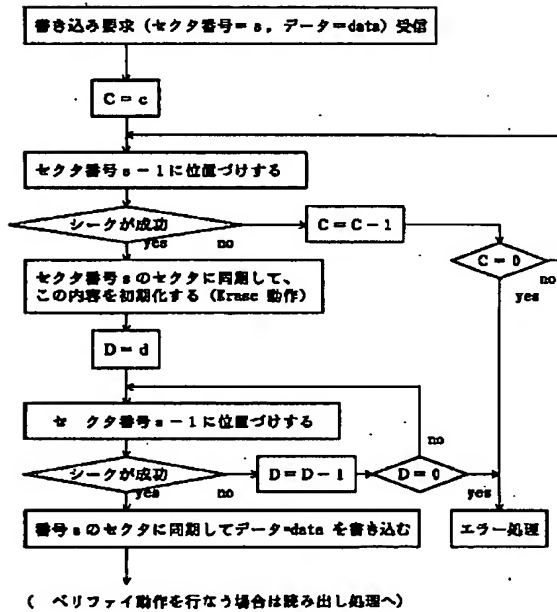
【図 2】

実施例 1 の構成例



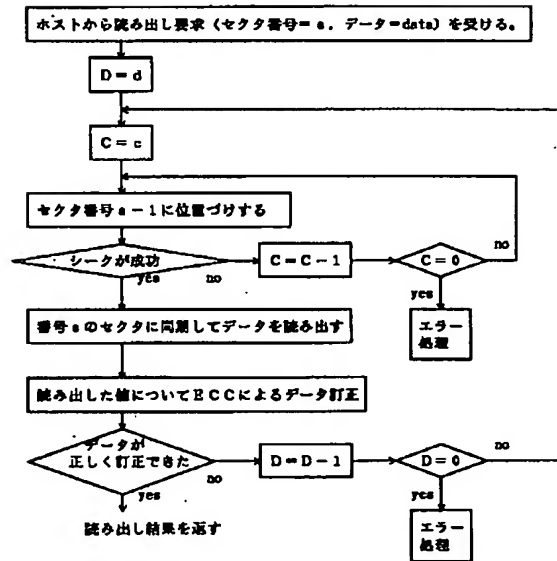
【図3】

実施例1のデータ書き込み処理フロー例



【図4】

実施例1のデータ読み出し処理フロー例



【図5】

エラー率に従ってエラー処理を行う疑似コード例

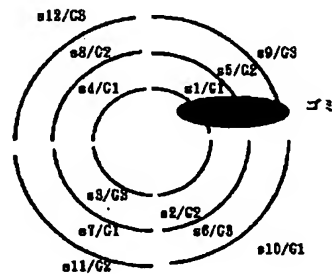
```

動作モードを「書き込み時検証なし」に設定。
HDD の保存すべきデータがある間、次を繰り返す。
{
  パリティバッファ PB[1]...PB[g] をゼロクリアする。
  パリティグループの大きさ n を既定値にする。
  j = 1
  while j <= n
  {
    エラー率計測手段から、エラー率に関する情報を受け取る
    エラー率にしたがって n を増減させる。
    if (エラー率が予め定めた E1 を超える)
      動作モードを「書き込み時検証あり」に設定。
    else if エラー率が予め定めた閾値 E0 を下回る
      動作モードを「書き込み時検証なし」に設定。

    for i=1 to g
      HDDから1セクタ分のデータを読み出して、IBへ保存。
      PB[i] ← PB[i] と IB から求めたパリティ値 (xor など)
      IB のデータを00ディスクに書き出す。
    next i
  }
  for i=1 to g
    PB[i] のデータを00ディスクに書き出す。
  next i
}
パリティセクタの場所を示す管理情報を書き出す。
  
```

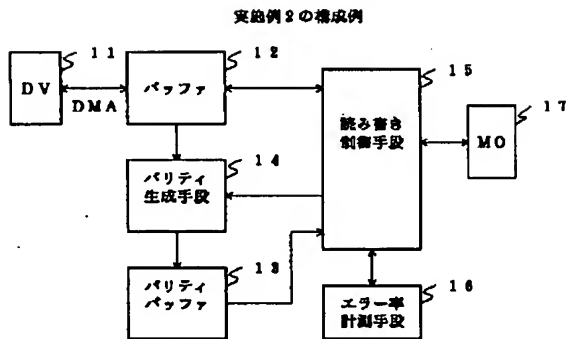
【図6】

実施例1のセクタ配置とゴミの付着例

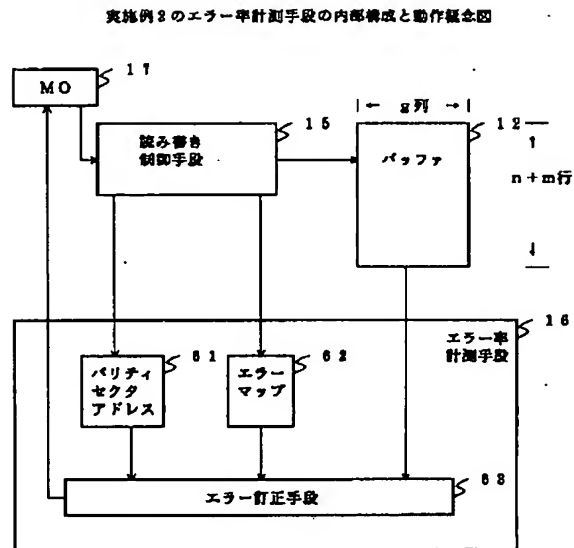


例 トラック当たりのセクタ数: $N=4$
 一つのパリティグループを構成するセクタ配置間隔: $k=8$
 セクタ番号: sxx
 パリティグループ番号: Cy
 パリティグループ1のセクタ: —

【図7】



【図8】



【図9】

エラー訂正処理例

```

for j = 1 to g
{
  バッファ中のj番目の列について、エラーマップ82に登録された
  セクタ数を調べる。
  そのセクタ数が1以上で、データ訂正能力の範囲内であれば
  ・パリティセクタのデータを用いてデータを訂正し、MOディスクに
    書き戻す。
  エラーマップ82に登録されたセクタ数が、パリティセクタによる
  訂正能力を上回るなら
  {
    ・対応するセクタをREADONLY命令（エラーを無視して読出す）を
      使って読み出し、一旦 バッファ12の当該部分に保存する。
      ECCデータ部分を別に保存する。
    ・バッファ12のデータについてパリティを求めて、
      検証とデータ訂正を試みる。
    （一般に読み出し不能となったセクタであっても全てのデータが
      読めないわけではなく、正しく読める部分も存在する。
      これで、読み出し不能であったセクタについてより正しい、
      あるいは正しい部分がより多いセクタデータが得られる。）
    ・読み出しエラーとなったセクタについて、ECCデータを含めて、
      セクタ内でのエラー訂正を行なう。
    ・エラー訂正したデータをMOディスクに書き戻す。
    ※ もともと読み出せていたセクタは害がない。
  }
}

```

【図10】

実施例3のディスク内の論理データのレイアウト例

ディスク全体の管理情報	
FAT	
ディレクトリ領域	
データ領域	
パリティセクタ領域①	32000 ~ 64000 セクタ程度（クラスター数で決まる） ☆クラスター毎のパリティデータ
二次パリティ管理②	管理部分は32-100セクタ程度
二次パリティセクタ領域	20000 ~ 100000 セクタ程度
エラー率管理領域③	100 セクタ程度 ☆ディスク内のリード/シークエラー多発部を管理

【図11】

二次パリティセクタ処理の疑似コード例

```

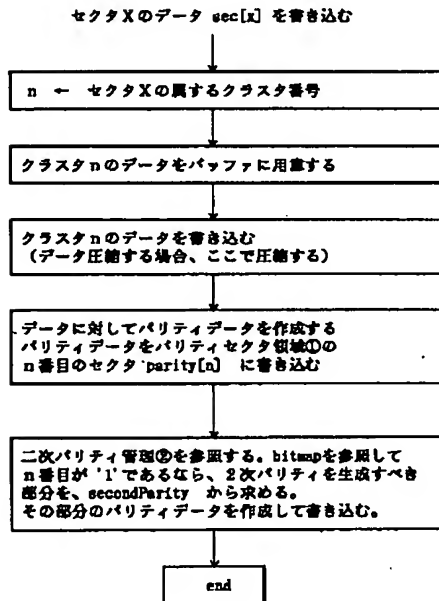
unsigned char bitmap[8192] ----- ①
struct {
  short   clust_no;
  unsigned pos;4; ----- ②
  unsigned num;4;
} secondParity[1024];

unsigned char
secondParitySector[1024] ----- ③

```

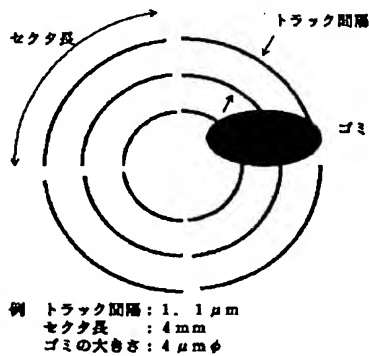
【図12】

実施例3のクラスタデータの書き込み処理フロー例



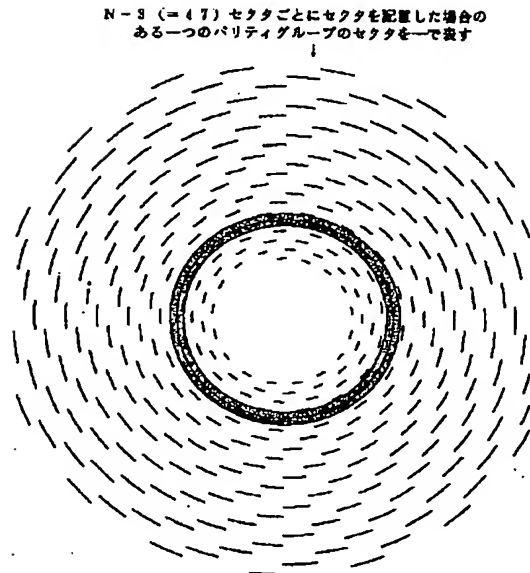
【図15】

同心円状にデータを記録するトラックとセクタ例



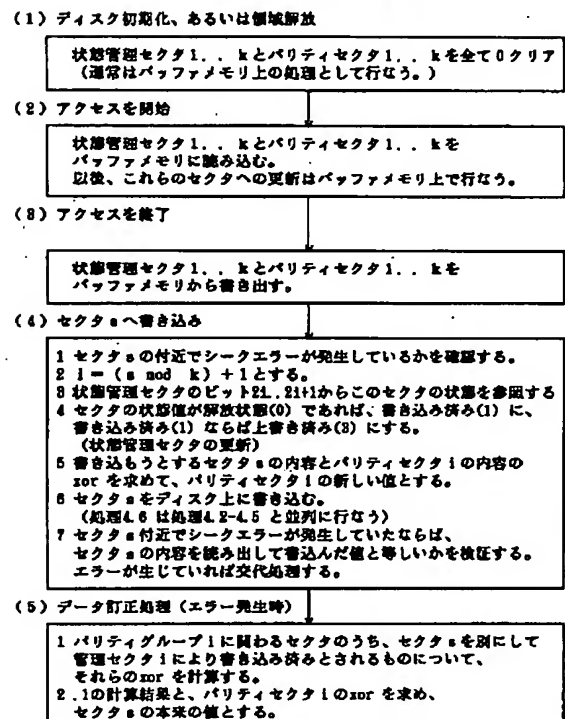
【図13】

実施例4のセクタ配置の概念図



【図14】

実施例4のデータ書き込み手順



【図16】

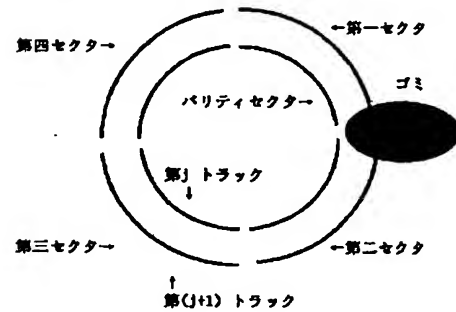
パリティセクタ生成例

第一セクタ	: \$00 \$08 \$0C \$33 \$45 \$AD \$51 \$20 \$5F
第二セクタ	: \$1A \$5E \$C0 \$53 \$B2 \$D8 \$97 \$4E \$29
第三セクタ	: \$DF \$49 \$C5 \$79 \$7E \$E3 \$D8 \$05 \$34
↓	
抽出したバイト列: \$00, \$1A, \$DF	
↓	
生成したパリティ符号 \$00 xor \$1A xor \$DF = \$35	
↓	
パリティセクタ	: \$35 \$17 \$DF \$99 \$89 \$90 \$35 \$82 \$52

\$:16進数を表す

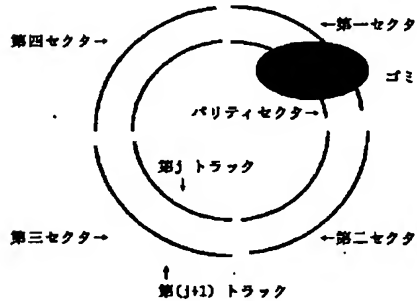
【図17】

連続してセクタを配置した従来例にゴミが付着した例1



【図18】

連続してセクタを配置した従来例にゴミが付着した例2



フロントページの続き

(51) Int. Cl. 7

G 1 1 B 20/18

識別記号

F I

G 1 1 B 20/18

テマコード(参考)

5 7 2 F